Equazioni di ricorrenza - Esercizi

Pietro Di Lena

Dipartimento di Informatica – Scienza e Ingegneria
Università di Bologna

Algoritmi e Strutture di Dati
Anno Accademico 2021/2022

Esercizio 1

- Risolvere la seguente equazione di ricorrenza

$$T(n) = \begin{cases} d & n \leq 1 \\ T(n/4) + c & n > 1 \end{cases}$$

con

  - Master Theorem
  - Metodo dell'iterazione
  - Metodo dell'albero di ricorsione
  - Metodo della sostituzione

Esercizio 1 - Soluzione

- Master Theorem

$$T(n) = \begin{cases} d & n \leq 1 \\ T(n/4) + c & n > 1 \end{cases}$$

Abbiamo $a = 1, b = 4, \alpha = \log_b a = 0, \beta = 0$

$$\alpha = \beta \Rightarrow T(n) = \Theta(n^0 \log n) = \Theta(\log n)$$

Esercizio 1 - Soluzione

- Metodo dell'iterazione

$$T(n) = \begin{cases} d & n \leq 1 \\ T(n/4) + c & n > 1 \end{cases}$$

Abbiamo

$$\begin{aligned} T(n) &= T(n/4) + c \\ &= T(n/4^2) + c + c \\ &= T(n/4^3) + c + c + c \\ &.. \\ &= T(n/4^i) + c \cdot i \end{aligned}$$

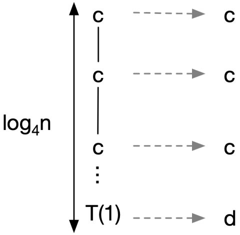
La ricorsione termina quando $n/4^i = 1 \Rightarrow i = \log_4 n$. Quindi

$$T(n) = T(1) + c \log_4 n = d + c \log_4 n = \Theta(\log n)$$

Esercizio 1 - Soluzione

- Metodo dell'albero di ricorsione

$$T(n) = \begin{cases} d & n \leq 1 \\ T(n/4) + c & n > 1 \end{cases}$$



- Abbiamo che

$$T(n) = c(\log_4 n - 1) + d = \Theta(\log n)$$

Esercizio 1 - Soluzione

- Metodo della sostituzione

$$T(n) = \begin{cases} d & n \leq 1 \\ T(n/4) + c & n > 1 \end{cases}$$

Ipotizziamo $T(n) = O(\log n)$, che implica ($\log = \log_2$)

$$\exists k > 0, \exists n_0 \geq 0 \text{ tale che } \forall n \geq n_0, T(n) \leq k \log n$$

1. Base: $T(4) = d + c \leq k \log 2$ vero per $k \geq d + c$, $n_0 = 4$
2. Induzione. Assumiamo che l'ipotesi sia vera per $T(n/4)$

$$\begin{aligned} T(n) &= T(n/4) + c \\ &\leq k \log(n/4) + c \\ &= k \log n - k \log 4 + c \\ &= k \log n - 2k + c \end{aligned}$$

Il passo induttivo è vero se esiste $k > 0$ tale che

$$k \log n - 2k + c \leq k \log n \Rightarrow k \geq c/2.$$

Concludiamo che $T(n) = O(\log n)$ (vera per $k \geq d + c$ e $n_0 = 4$)

Esercizio 2

- Risolvere la seguente equazione di ricorrenza

$$T(n) = \begin{cases} d & n \leq 1 \\ 2T(n/4) + c & n > 1 \end{cases}$$

con

- Master Theorem
- Metodo dell'iterazione
- Metodo dell'albero di ricorsione
- Metodo della sostituzione

Esercizio 2 - Soluzione

- Master Theorem

$$T(n) = \begin{cases} d & n \leq 1 \\ 2T(n/4) + c & n > 1 \end{cases}$$

Abbiamo $a = 1, b = 4, \alpha = \log_4 2 = 1/2, beta = 0$

$$\alpha > \beta \Rightarrow T(n) = \Theta(n^\alpha) = \Theta(\sqrt{n})$$

Esercizio 2 - Soluzione

- Metodo dell'iterazione

$$T(n) = \begin{cases} d & n \leq 1 \\ 2T(n/4) + c & n > 1 \end{cases}$$

Abbiamo

$$\begin{aligned} T(n) &= 2T(n/4) + c \\ &= 2^2 T(n/4^2) + 2c + c \\ &= 2^3 T(n/4^3) + 2^2 c + 2c + c \\ &.. \\ &= 2^i T(n/4^i) + \sum_{k=0}^{i-1} 2^k c \end{aligned}$$

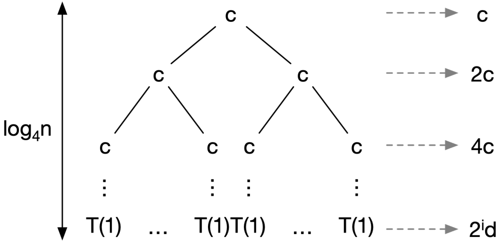
La ricorsione termina quando $n/4^i = 1 \Rightarrow i = \log_4 n$. Quindi

$$T(n) = 2^{\log_4 n} T(1) + c\frac{2^{\log_4 n} - 1}{2-1} = \sqrt{n}d + c(\sqrt{n} - 1) = \Theta(\sqrt{n})$$

Esercizio 2 - Soluzione

- Metodo dell'albero di ricorsione

$$T(n) = \begin{cases} d & n \leq 1 \\ 2T(n/4) + c & n > 1 \end{cases}$$



- Abbiamo che

$$T(n) = c \sum_{i=0}^{\log_4 n - 1} 2^i + 2^{\log_4 n} d = c(\sqrt{n} - 1) + d\sqrt{n} = \Theta(\sqrt{n})$$

Esercizio 2 - Soluzione

- Metodo della sostituzione

$$T(n) = \begin{cases} d & n \leq 1 \\ 2T(n/4) + c & n > 1 \end{cases}$$

Ipotizziamo $T(n) = \Omega(\sqrt{n})$ (difficile dimostrare $O(\sqrt{n})$), che implica

$$\exists k > 0, \exists n_0 \geq 0 \text{ tale che } \forall n \geq n_0, T(n) \geq k\sqrt{n}$$

1. Base: $T(1) = d \geq k\sqrt{1}$ vero per ogni $k \leq d, n_0 = 1$
2. Induzione. Assumiamo che l'ipotesi sia vera per $T(n/4)$

$$\begin{aligned} T(n) &= 2T(n/4) + c \\ &\geq 2k\sqrt{n/4} + c \\ &= k\sqrt{n} + c \\ &\geq k\sqrt{n} \end{aligned}$$

Concludiamo che $T(n) = \Omega(\sqrt{n})$ (vera per $0 < k \leq d$ e $n_0 = 1$)

Esercizio 3

- Risolvere la seguente equazione di ricorrenza

$$T(n) = \begin{cases} d & n \leq 1 \\ 4T(n/4) + c & n > 1 \end{cases}$$

con

  - Master Theorem
  - Metodo dell'iterazione
  - Metodo dell'albero di ricorsione
  - Metodo della sostituzione

Esercizio 3 - Soluzione

- Master Theorem

$$T(n) = \begin{cases} d & n \leq 1 \\ 4T(n/4) + c & n > 1 \end{cases}$$

Abbiamo $a = 1, b = 4, alpha = \log_4 4 = 1, \beta = 0$

$$\alpha > \beta \Rightarrow T(n) = \Theta(n^{\alpha}) = \Theta(n)$$

Esercizio 3 - Soluzione

- Metodo dell'iterazione

$$T(n) = \begin{cases} d & n \leq 1 \\ 4T(n/4) + c & n > 1 \end{cases}$$

Abbiamo

$$\begin{aligned} T(n) &= 4T(n/4) + c \\ &= 4^2 T(n/4^2) + 4c + c \\ &= 4^3 T(n/4^3) + 4^2 c + 4c + c \\ &.. \\ &= 4^i T(n/4^i) + \sum_{k=0}^{i-1} 4^k c \end{aligned}$$

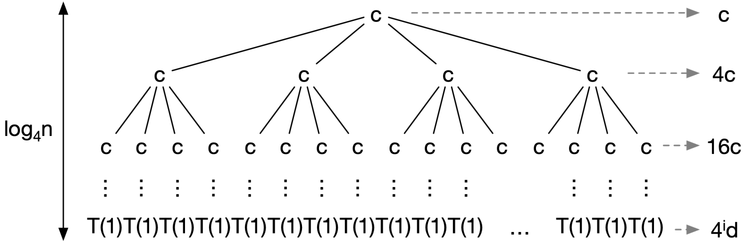
La ricorsione termina quando $n/4^i = 1 \Rightarrow i = \log_4 n$. Quindi

$$T(n) = 4^{\log_4 n} T(1) + c \frac{4^{\log_4 n} - 1}{2 - 1} = nd + c(n-1) = \Theta(n)$$

Esercizio 3 - Soluzione

- Metodo dell'albero di ricorsione

$$T(n) = \begin{cases} d & n \leq 1 \\ 4T(n/4) + c & n > 1 \end{cases}$$



- Abbiamo che

$$T(n) = c \sum_{i=0}^{\log_4 n - 1} 4^i + 4^{\log_4 n} d = c(n-1) + dn = \Theta(n)$$

Esercizio 3 - Soluzione

- Metodo della sostituzione

$$T(n) = \begin{cases} d & n \leq 1 \\ 4T(n/4) + c & n > 1 \end{cases}$$

Ipotizziamo $T(n) = \Omega(n)$ (difficile dimostrare $O(n)$), che implica

$$\exists k > 0, \exists n_0 \geq 0 \text{ tale che } \forall n \geq n_0, T(n) \geq kn$$

1. Base: $T(1) = d \geq k \cdot 1$ vero per ogni $k \leq d, n_0 = 1$
2. Induzione. Assumiamo che l'ipotesi sia vera per $T(n/4)$

$$\begin{aligned} T(n) &= 4T(n/4) + c \\ &\geq 4k(n/4) + c \\ &= kn + c \\ &\geq kn \end{aligned}$$

Concludiamo che $T(n) = \Omega(n)$ (vera per $0 < k \leq d$ e $n_0 = 1$)

Esercizio 4

- Risolvere la seguente equazione di ricorrenza

$$T(n) = \begin{cases} d & n \leq 1 \\ 8T(n/4) + c & n > 1 \end{cases}$$

con

  - Master Theorem
  - Metodo dell'iterazione
  - Metodo dell'albero di ricorsione
  - Metodo della sostituzione

Esercizio 4 - Soluzione

- Master Theorem

$$T(n) = \begin{cases} d & n \leq 1 \\ 8T(n/4) + c & n > 1 \end{cases}$$

Abbiamo $a = 1, b = 4, \alpha = \log_4 8 = 3/2, \beta = 0$

$$\alpha > \beta \Rightarrow T(n) = \Theta(n^{3/2}) = \Theta(\sqrt{n^3})$$

Esercizio 4 - Soluzione

- Metodo dell'iterazione

$$T(n) = \begin{cases} d & n \leq 1 \\ 8T(n/4) + c & n > 1 \end{cases}$$

Abbiamo

$$\begin{aligned} T(n) &= 8T(n/4) + c \\ &= 8^2 T(n/4^2) + 8c + c \\ &= 8^3 T(n/4^3) + 8^2 c + 8c + c \\ &.. \\ &= 8^i T(n/4^i) + \sum_{k=0}^{i-1} 8^k c \end{aligned}$$

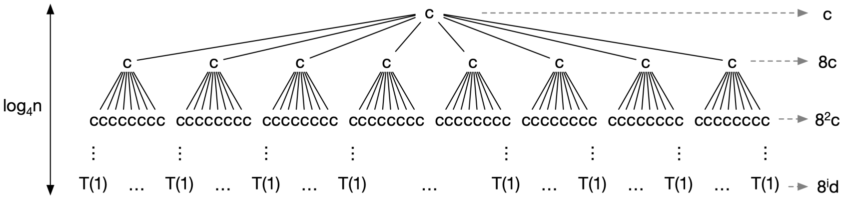
La ricorsione termina quando $n/4^i = 1 \Rightarrow i = \log_4 n$. Quindi

$$T(n) = 8^{\log_4 n} T(1) + c\frac{8^{\log_4 n} - 1}{2 - 1} = \sqrt{n^3}d + c(\sqrt{n^3} - 1) = \Theta(\sqrt{n^3})$$

Esercizio 4 - Soluzione

- Metodo dell'albero di ricorsione

$$T(n) = \begin{cases} d & n \leq 1 \\ 8T(n/4) + c & n > 1 \end{cases}$$



- Abbiamo che

$$T(n) = c \sum_{i=0}^{\log_4 n - 1} 8^i + 8^{\log_4 n} d = c(\sqrt{n^3} - 1) + d\sqrt{n^3} = \Theta(\sqrt{n^3})$$

Esercizio 4 - Soluzione

- Metodo della sostituzione

$$T(n) = \begin{cases} d & n \leq 1 \\ 8T(n/4) + c & n > 1 \end{cases}$$

Ipotizziamo $T(n) = O(n^2)$ (difficile dimostrare $O(\sqrt{n^3})$), che implica

$$\exists k > 0, \exists n_0 \geq 0 \text{ tale che } \forall n \geq n_0, T(n) \leq kn^2$$

1. Base: $T(1) = d \geq k \cdot 1$ vero per ogni $k \geq d, n_0 = 1$
2. Induzione. Assumiamo che l'ipotesi sia vera per $T(n/4)$

$$\begin{aligned} T(n) &= 8T(n/4) + c \\ &\leq 8k(n^2/16) + c \\ &= k(n^2/2) + c \end{aligned}$$

Il passo induttivo è vero se esiste $k > 0$ tale che

$$k(n^2/2) + c \leq kn^2 \Rightarrow kn^2 \geq 2c \Rightarrow k \geq \frac{2c}{n^2}$$

Concludiamo che $T(n) = O(n^2)$ ($k \geq \max\{2c, d\}$ e $n_0 = 1$)